XML in WebObjects

Session 704

XML in WebObjects

Bob Frank
Senior Consulting Engineering

Mike Brumbelow
Technical Architech

Han Ming Ong
WebObjects Engineering

What You Will Learn

- Key Concepts of XML
- XML Serialization of EO's
- XSLT in WebObjects
- XML-RPC
    - What is it
    - How to use it
    - Compare with SOAP
- Count the letter 'X'

Key Concepts of XML

Han Ming Ong
WebObjects Engineering

Key Concepts of XML

- What is XML?
- Constraints
    - Document Type Definition
    - Schema
- Processing
    - DOM, SAX, JAXP
    - Parsers

What Is XML?

- Structured data
  - Hierarchical, tree-like form
  - Great for developers
  - Familiar for layman
- Portable form of data
  - Text, accepts Unicode characters
  - Allows encoding specification

Example

```
<?xml version="1.0" encoding="UTF-8" ?>

<purchaseOrder orderDate="2002ƒl2‘ł4»">
  <comment>«lƒ„Ω¯øl“ªµ„ <comment>
  <item partNum="872-AA">
    <productName>Lawnmower</productName>
    <quantity>1</quantity>
    <USPrice>148.95</USPrice>
  </item>
</purchaseOrder>
```

Constraints

- Well-formedness—XML 1.0 specification
- Validity
    - Contract of structure
    - Domain specific
- Document Type Definition
- Schema

DTD

- Document Type Definition
- ‘Weird’ syntax
- No typing facility
- Namespace
    - Avoid name conflict
    - Package in Java

Example

```
<!ELEMENT purchaseOrder (comment, item)>
<!ATTLIST purchaseOrder
        orderDate  PCDATA #REQUIRED
>

<!ELEMENT comment (#PCDATA)>
...
```

Schema

- Same goal as DTD
- Written in XML format
- Data Typing
- Namespace

Example

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.apple.com/publications"
        xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:apple="http://www.apple.com/publications">

  <element name="apple:purchaseOrder">
    <complexType>
      <sequence>
        <element name=" apple:comment" type="string" />
        <element name=" apple:item" type="apple:ItemType" />
      </sequence>
    </complexType>
    ...
```

What XML Is Not

- Complete interoperability
  - Still have to talk common language
- Automatically used
  - Got to find domain to use it
  - Content publishing, storing configuration info
  - Web services

DOM

- Document Object Model
- Very general tree structure
- Better used for producing XML data
- Memory intensive

SAX

- Simple API for XML
- Faster than DOM parsing
- Sequential parsing
- Standard callbacks

JAXP

- Neither SAX nor DOM
- Purely abstract layer for both
- Allows any parser/transformer to easily plug in
    - Vendor independent

Parsers

- Apache Xerces
  http://xml.apache.org
- Sun Crimson
  http://xml.apache.org/crimson
- IBM XML4J
- James Clark XP

Integration With WO

- JavaXML.framework
  - Xerces 2 parser
  - Xalan 2 transformer
- XML serialization
  - WOXMLCoder, WOXMLDecoder
  - NSXMLInputStream, NSXMLOutputStream

WOXMLCoder and Decoder

- They are deprecated
  - Only use for legacy apps
  - Will still be around
- GOOD BYE Non-standard Mapping File
- HELL0 XSLT

XML Serialization

- Easy
  - **Implements Serializable**
- NSXMLInputStream
  - **extends ObjectInputStream**
- NSXMLOutputStream
  - **extends ObjectOutputStream**
- WebObject does the heavy lifting
- Industry standard

Default Format

```
<?xml version="1.0" encoding="UTF-8"?>

<content
   xmlns="http://www.apple.com/webobjects/XMLSerialization">

   <object id="2">
      <class flag="2" id="0"  name="Person" suid="1001">
         <field name=”firstName" type="java.lang.String"/>
      </class>

      <string field="firstName" id="3">Mike</string>
   </object>

</content>
```

Example—Writing

```
NSXMLOutStream xmlOut = new NSXMLOutputStream(outStream);

xmlOut.writeObject( myObject );

xmlOut.close();
```

Example—Reading

```
NSXMLInputStream xmlIn = new NSXMLInputStream(inStream);

myRestoredObject = (MyClass) xmlIn.readObject();

xmlIn.close();
```

Demo

XML Based Configuration Files

Han Ming Ong

EO XML Serialization and XSLT

Mike Brumbelow
Technical Architect

EO XML Serialization

- Good news . . easy as pie!
  - xmlOut.writeObject(myEO);
  - myEO = (MyEOClass) xmlIn.readObject();
- Managing your object graph
  - Not much to worry about—does not fire faults

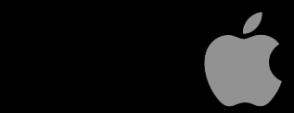
Demo

RSS and EO XML Serialization

Mike Brumbelow

XSL—eXtensible Stylesheet Language

- Two rule based vocabularies: XSLT and XSL FO
- Transform source XML documents
- Specify the formatting of XML documents
- Formatting Objects (XSL FO)
    - An XML vocabulary that defines page layout
    - Binary output, PDF and WORD
- See: **http://xml.apache.org/fop**



XSLT—Rules Document of XSL

- Extracting data from source XML document
- The vocabulary has constructs for
    - Template matching
    - Copying
    - Control structures
    - Iteration
    - Maintaining state
- Locate data using XPath expressions

XSLT—Components

- A valid XSL Transformation document
- Apache Xalan XSL Transformation parser
  - Ship with WO 

- A valid source XML document
- Output XML document

Example

- Foo.xml

```
<?xml version="1.0"?>
<doc>Hello</doc>
```

- Foo.xslt

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    version="1.0">
<xsl:template match="doc">
<out><xsl:value-of select="."/></out>
</xsl:template>
</xsl:stylesheet>
```

- Foo.out

```
<?xml version="1.0"?>
<out>Hello</out>
```

Other Uses of XSLT

- Allows transformation of XML to
  - HTML
  - XHTML
  - WML
  - PDF
  - Etc.

XSLT Processors

- Apache Xalan
  http://xml.apache.org/xalan-j/index.html
- SourceForge Saxon
  http://saxon.sourceforge.net/

Demo

XSL Transformer

Mike Brumbelow

WOXMLNode and XML-RPC

Bob Frank
Senior Consulting Engineering

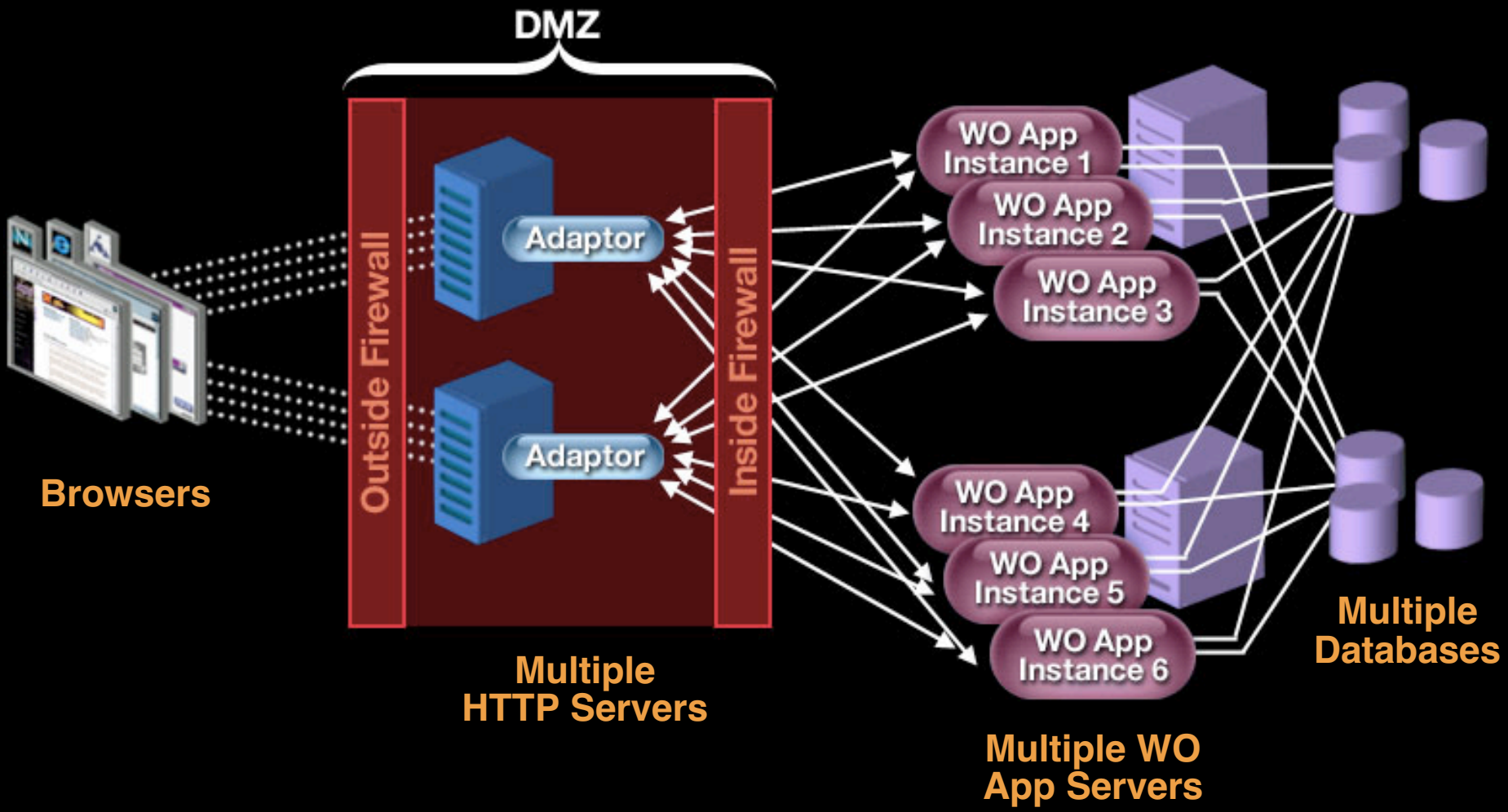
WOXMLNode and WOBuilder

- Graphical way to create XML Documents
  - Tight Integration with WOBuilder
- WOXMLNode
  - Document format under your precise control
  - Probably more compact representation

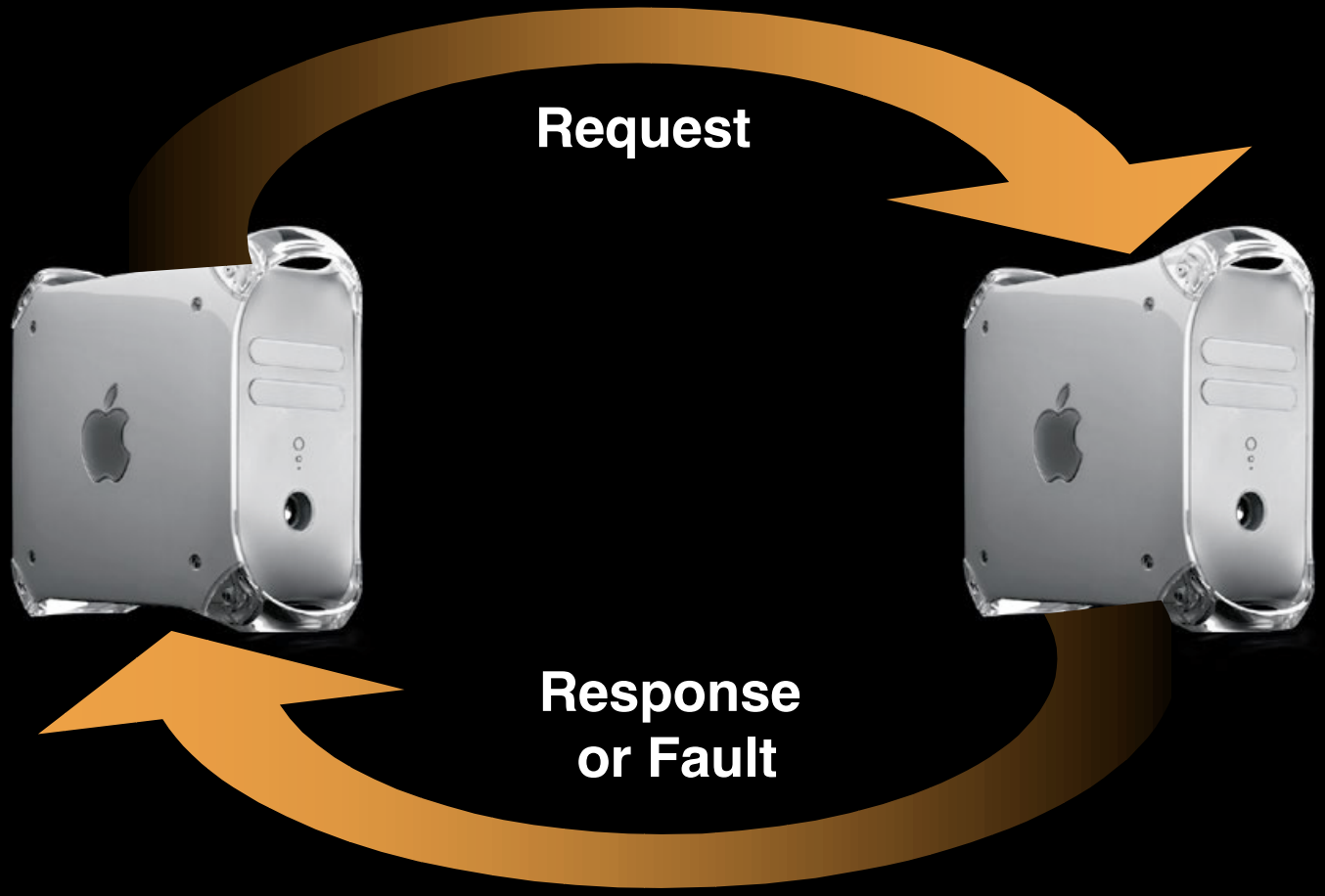
Browsers—Mime Header

```
//  You should set the mime type to “text/xml”

public void appendToResponse(WOResponse
response, WOContext context) {
        response.setHeader("text/xml", "content-type");
        super.appendToResponse(response, context);

}
```

Typical WO HTTP Application



XML-RPC Communication



XML-RPC and SOAP

- XML-RPC = Remote Procedure Calls
  - Parameter marshalling mechanism
  - Simpler than SOAP
  - Verbose
- SOAP = Simple Object Access Protocol
  - More typing specifications
  - More compact
  - See Session 705—Web Services

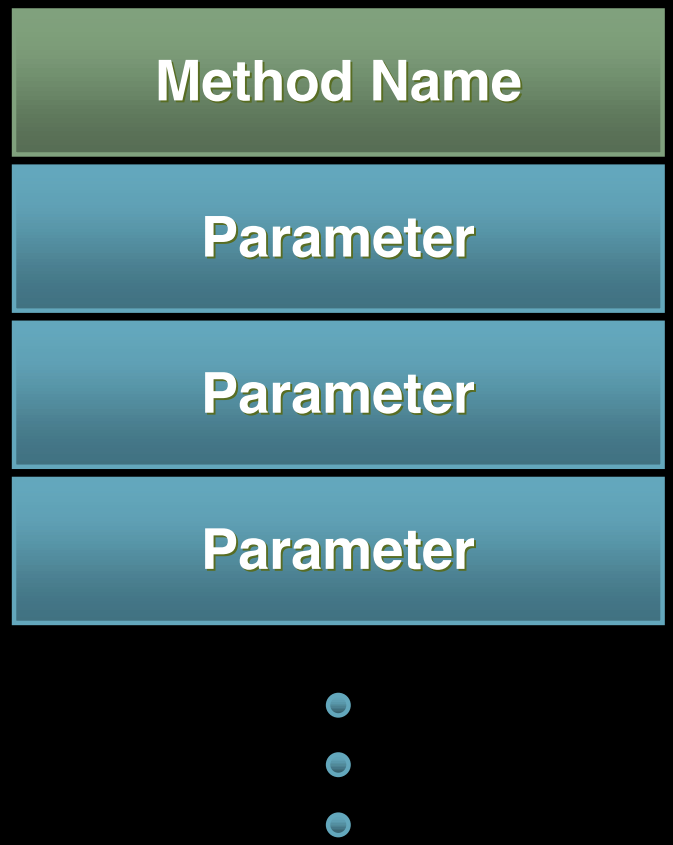
XML-RPC and SOAP (Cont.)

- SOAP is W3C Note
- Both address remote method invocation
    - No screen scraping
- Neither addresses
    - Object Activation
    - Garbage Collection

XML-RPC Internals

- Strict “RPC” concept
- Requests are ‘methodCall’s’
- Responses are ‘methodResponse’s’
    - Uses faults for error conditions
    - Similar to Exceptions

Method Call



XML-RPC Types

- boolean
- double
- dateTime.iso8601
- int (signed, 32-bit)
- string
    - May be empty
    - ASCII/UniCode
    - Default type

XML-RPC Types

- base64
    - binary data
- array
- struct (dictionary)
    - Names are strings (i.e. keys)
    - Values may be of any type
    - Unordered

XML-RPC Types

- Values are typed with type tags inside ‘value’ tag:
  - **<int></int>**
  - **<boolean></boolean>**
  - **<string></string>**
  - **<double></double>**
  - **<dateTime.iso8601></dateTime.iso8601>**
  - **<base64></base64>**
- All values wrapped with ‘value’ tag
  - **<value></value>**
  - **<value><string>Hello World!</string></value>**

Arrays

- Ordered list of ‘values’
  - **<array></array>**
  - **<data></data>**

```
<array>
  <data>
    <value><int>89567</int></value>
    <value><string>Canada</string></value>
    <value><boolean>0</boolean></value>
    <value><int>-31</int></value>
    <value><double>3.14159265358979323846264433</double></value>
  </data>
</array>
```

Structs = Dictionaries

- Unordered list of ‘members’ (key-value pairs)
  - **<struct></struct>**
  - **<member></member>**
    - **<name></name>**
    - **<value></value>**

```
<struct>
  <member>
    <name>lowerBound</name>
    <value><int>18</int></value>
  </member>
  <member>
    <name>upperBound</name>
    <value><int>139</int></value>
  </member>
</struct>
```

XML-RPC Request

```
POST /RPC2 HTTP/1.0
User-Agent: BobsApp/5.1.2 (MacOSX)
Host: xmlrpc.bob.apple.com
Content-Type: text/xml
Content-length: 181

<?xml version="1.0"?>
<methodCall>
   <methodName>examples.getStateName</methodName>
   <params>
      <param>
         <value><int>32</int></value>
      </param>
   </params>
</methodCall>
```

XML-RPC Response

```
HTTP/1.1 200 OK
Connection: close
Content-Length: 158
Content-Type: text/xml
Date: Fri, 17 Jul 2001 19:55:08 GMT
Server: Apple WebObjects/5.1.2-Darwin

<?xml version="1.0"?>
<methodResponse>
  <params>
    <param>
      <value><string>Minnesota</string></value>
    </param>
  </params>
</methodResponse>
```

XML-RPC Fault

```
<?xml version="1.0"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>23</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string>Unknown Stock Symbol</string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>
```

Common Server Extensions

- system.multicall (Box carring)
  - Multiple method calls in a single message
  - Very good for combining small messages
- system.listMethods
- system.methodSignature
- system.methodHelp

Other Key Features

- Easy to transport over HTTPS
- Supports HTTP user/password authentication
    - More secure with HTTPS

Demo

XML-RPC and Blogging

Bob Frank

Why XML-RPC

- "Worse is Better"
  - Only tries to do 1 thing well (RPC)
  - Easy to add new methods on server
- Spec is 19K
  - **www.xmlrpc.org/spec**
  - SOAP Spec = 121K

WebObjects Lab

- Located downstairs in Room L
- Lab hours
  - Monday 12:00pm–6:00pm
  - Tuesday 9:00am–2:00pm*
  - Wednesday 9:00am–6:00pm
  - Thursday 9:00am–6:00pm
  - Friday 9:00am–6:00pm

*Conversion Workshop Tuesday 2-6pm. Sign up in Lab

Roadmap

405 Java Web Services	Room C **Wed., 10:30am**
705 WebObjects and Web Services	Room A1 **Wed., 2:00pm**
713 WebObjects Solutions	Room A1 **Fri., 9:00am**
FF013 WebObjects	Room A1 **Fri., 3:30pm**

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

Apple Professional Services (Training, Support, Consulting)
(800) 848-6398
services@apple.com

http://developer.apple.com/wwdc2002/urls.html

For More Information

- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  www.apple.com/services/technicalsupport
- Other places
  - **www.apple.com/webobjects**
  - **developer.apple.com/webobjects**
  - **www.apple.com/services**
  - **www.info.apple.com/webobjects**
- Subscribe
  webobjects-announce@apple.com

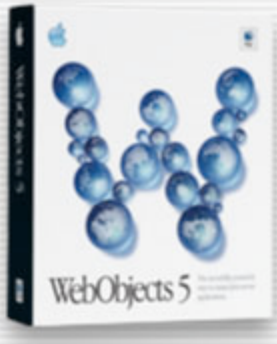
Documentation

Data Serialization Using XML (forthcoming)

How to Access Documentation

- Most up-to-date: PDF and HTML
  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand
  Vervante.com under Related Resources
- Product CD
  Documents folder and installed in /Developer/Documentation/WebObjects
- In the box (localized)
  Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications, Discovering WebObjects for HTML
- Check ADC News for latest updates
  http://developer.apple.com/devnews

Q&A



Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html


WWDC2002


WWDC2002


WWDC2002